Effective Roughness Models

Andrew Whelan

Contents

N Notation

$(\vec{r}, t) \in \mathbb{R}^{3+1}$	Coordinates
$\hat{\mathbf{e}} = (\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3)$	Abstract vector space basis
$\Sigma_\partial = \{\vec{r} \mid f_{\Sigma_\partial}(\vec{r}) = 0\}$	Boundary between two media
$\Sigma_R = \{\vec{r} \mid f_{\Sigma_R}(\vec{r}) = 0\}$	Surface of point-receivers
$\Omega = \{\vec{r} \mid f_{\Sigma_\partial}(\vec{r}) < 0\}$	Reflected volume
$\vec{r}_T, \vec{r}_\partial, \vec{r}_R$	Positions of transmitter, scattering point, and receiver
$\hat{n}_\Sigma(\vec{r}) = \frac{\nabla f_\Sigma}{\lvert\nabla f_\Sigma\rvert}$	Unit surface normal along Σ
$d\vec{A}_\Sigma(\vec{r}) = \hat{n}_\Sigma(\vec{r})\ dA_\Sigma(\vec{r})$	Vector surface area element of Σ
$\mathcal{E}(\vec{r}, t) = \mathrm{Re}\{e^{j\omega t}\mathbf{E}(\vec{r})\}$	Electric Field
$\mathbf{E}(\vec{r}) \in \mathbb{C}^3$	Electric Field Phasor
$\mathcal{H}(\vec{r}, t) = \mathrm{Re}\{e^{j\omega t}\mathbf{H}(\vec{r})\}$	Magnetic Field
$\mathbf{H}(\vec{r}) \in \mathbb{C}^3$	Magnetic Field Phasor
$\vec{\mathcal{S}}(\vec{r}, t) \stackrel{\text{def}}{=} \mathcal{E} \times \mathcal{H}$	Poynting vector
$\langle\vec{\mathcal{S}}\rangle(\vec{r}) \stackrel{\text{def}}{=} \frac{\omega}{2\pi}\int_0^{\frac{2\pi}{\omega}} \vec{\mathcal{S}}(t)dt$	Time-averaged Poynting vector
$\vec{\xi}_i \stackrel{\text{def}}{=} (\epsilon_i, \mu_i, \sigma_i)$	Constitutive Parameters (i indexes the medium)
$\eta \stackrel{\text{def}}{=} \sqrt{\frac{j\omega\mu}{\sigma + j\omega\epsilon}} \in \mathbb{C}$	Intrinsic impedance of wave medium
$\mathbf{E}(\vec{r}) = \lvert\mathbf{E}(\vec{r})\rvert e^{j\chi_E}\hat{\mathbf{e}}_{\chi_E}$	Linearly polarized **E** phase decomposition
$\chi_E,\ \chi_H \in \mathbb{R}$	Propagation phases
$\hat{\mathbf{e}}_{\chi_E},\ \hat{\mathbf{e}}_{\chi_H} \in \mathbb{R}^3$	Unit (linear) polarization vectors
$\phi \stackrel{\text{def}}{=} \cos^{-1}(\hat{\mathbf{e}}_{\chi_E} \cdot \hat{\mathbf{e}}_{\chi_H})$	Spatial phase
$\theta_\Sigma \stackrel{\text{def}}{=} \cos^{-1}(\frac{\langle\vec{\mathcal{S}}\rangle \cdot \hat{n}_\Sigma}{\lvert\langle\vec{\mathcal{S}}\rangle\rvert})$	Propagation angle
$dP_\Sigma(\vec{r}) \stackrel{\text{def}}{=} \langle\vec{\mathcal{S}}\rangle \cdot d\vec{A}_\Sigma$	Incremental power over Σ
$\mathcal{T}(\vec{r}) + \mathcal{R}(\vec{r}) = 1$	Local Power Balance
$\mathcal{T}(\vec{r})$	Transmittance (portion of dP transmitted through Σ)
$\mathcal{R}(\vec{r}) = \mathcal{R}_S(\vec{r}) + \mathcal{R}_R(\vec{r})$	Reflectance (portion of dP reflected back)
$\mathcal{R}_R(\vec{r}) = R^2\lvert\Gamma(\vec{r})\rvert^2$	Specular reflectance
$\mathcal{R}_S(\vec{r}) = S^2\lvert\Gamma(\vec{r})\rvert^2$	Diffuse reflectance
$\Gamma(\vec{r}) \in \mathbb{C}$	Fresnel reflection coefficient

A Model Assumptions

A.1 Scattering Parameter S

$$S \in [0,1], \text{ is a constant parameter capturing } \textit{effective roughness} \text{ of } \Sigma_\partial. \quad \text{(A.1)}$$

A.2 Transmittance is weakly dependent on Geometry

$$\text{We can vary } R \text{ and } S \text{ without adversely affecting } \mathcal{T}: \quad \frac{\partial \mathcal{T}}{\partial(R,S)} \approx 0 \quad \text{(A.2)}$$

A.3 Homogeneous Scatterer

$$S^2|\Gamma|^2 dP_{\Sigma_\partial} = \iint_\Omega I(\theta,\phi) d\theta d\phi \quad \text{(A.3)}$$

A.4 Incoherent Scattering

Scattered wave phases are random & uncorrelated, so power sums incoherently:

$$\begin{aligned} S^2 dP_{\Sigma_{R,S}}(\vec{r}) &\stackrel{\text{def}}{=} dP_{\Sigma_R} - R^2 dP_{\Sigma_{R,GO}} \\ &= \int_{\Sigma_\partial} \left(\int_{\theta_{\Sigma_R} - \frac{d\theta_{\Sigma_R}}{2}}^{\theta_{\Sigma_R} + \frac{d\theta_{\Sigma_R}}{2}} \int_{\phi_{\Sigma_R} - \frac{d\phi_{\Sigma_R}}{2}}^{\phi_{\Sigma_R} + \frac{d\phi_{\Sigma_R}}{2}} I(\theta,\phi)\ d\phi\ d\theta \right) \end{aligned} \quad \text{(A.4)}$$

F Field Assumptions

F.1 Remote Antennas

Antennas are not in the near-field (an outdoor scenario), so we have

$$\phi = \frac{\pi}{2}, \quad \chi_E = \chi_H \quad \text{(F.1)}$$

F.2 Medium 1

The wave is travelling in free space until it hits an obstruction:

$$(\epsilon_1, \mu_1, \sigma_1) = (\epsilon_0, \mu_0, \sigma_0). \quad \text{(F.2)}$$

F.3 Medium 2

The obstruction is a PEC (this can be adjusted later):

$$(\epsilon_2, \mu_2, \sigma_2) = (\epsilon_0, \mu_0, \infty). \tag{F.3}$$

D Direct Consequences

D.1 Local Incremental Power

The formula for local incremental power can be simplified into two useful forms:

$$\begin{aligned} dP_\Sigma(\vec{r}) &= \langle \vec{\mathcal{S}}(\vec{r}) \rangle \cdot d\vec{A}_\Sigma(\vec{r}) \\ &= \mathrm{Re}(\mathbf{E}(\vec{r}) \times \mathbf{H}(\vec{r})^*) \cdot \hat{\mathbf{n}}_\Sigma(\vec{r}) \quad \text{(D.1a)} \\ &= |\langle \vec{\mathcal{S}} \rangle| \cos(\theta_\Sigma) dA_\Sigma \quad \text{(D.1b)} \end{aligned}$$

D.2 Transmittance Approximation

(A.1) and (A.2) mean that we can adjust S and R without adversely affecting T. So, considering the perfectly specular case $S = 0, R = 1$, we get

$$\begin{aligned} &\mathcal{T} \approx 1 - |\Gamma|^2 \\ \implies\ & R^2 + S^2 \approx 1 \end{aligned} \tag{D.2}$$

1 2D Lambertian-Model

Here, we have a restricted 2d setup, and assume a cylindrical wave source:

$$|\langle \vec{\mathcal{S}} \rangle| = \frac{K}{|\vec{r} - \vec{r}_T|} \tag{1.1}$$

We can calculate K by integrating over a wavefront that doesn't intersect Σ_∂:

$$\begin{aligned} P_0 &= \int_0^{2\pi} |\langle \vec{\mathcal{S}} \rangle| dl = \int_0^{2\pi} \frac{K}{|\vec{r} - \vec{r}_T|} dl \\ &= \int_0^{2\pi} \frac{K}{|\vec{r} - \vec{r}_T|} |\vec{r} - \vec{r}_T| d\theta = 2\pi K \\ \overset{(1.1),(D.1b)}{\implies} dP_{\Sigma_\partial}(\vec{r}) &= \frac{P_0 \cos\theta_{\Sigma_\partial}}{2\pi |\vec{r} - \vec{r}_T|} dl_{\Sigma_\partial} \quad \text{(1.2)} \end{aligned}$$

The Lambertian scattering assumption in 2D is (see (A.3)):

$$
\begin{aligned}
I(\theta) &= D\cos\theta && (1.3)\\
\implies S^2|\Gamma|^2 dP_{\Sigma_\partial} &= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} D\cos\theta d\theta = 2D \\
\overset{(1.3),(1.2)}{\implies} I(\theta) &= \frac{S^2|\Gamma|^2 P_0 \cos\theta_{\Sigma_\partial} dl_{\Sigma_\partial}}{4\pi|\vec{r}-\vec{r}_T|}\cos\theta \\
\overset{(A.4)}{\implies} dP_{\Sigma_{R,S}} &= \frac{1}{S^2}\int_{\Sigma_\partial}\left(\int_{\theta_{\Sigma_R}-\frac{d\theta_{\Sigma_R}}{2}}^{\theta_{\Sigma_R}+\frac{d\theta_{\Sigma_R}}{2}} \frac{P_0 S^2|\Gamma|^2}{4\pi|\vec{r}_\partial-\vec{r}_T|}\cos\theta d\theta\right)\cos\theta_{\Sigma_\partial} dl_{\Sigma_\partial} \\
&= \int_{\Sigma_\partial}\left(\frac{P_0|\Gamma|^2\cos\theta_{\Sigma_R}}{4\pi|\vec{r}_\partial-\vec{r}_T|}d\theta_{\Sigma_R}\right)\cos\theta_{\Sigma_\partial} dl_{\Sigma_\partial} \\
&= \frac{P_0}{4\pi}\left(\int_{\Sigma_\partial}\frac{|\Gamma|^2\cos\theta_{\Sigma_R}\cos\theta_{\Sigma_\partial}}{|\vec{r}_\partial-\vec{r}_T||\vec{r}_R-\vec{r}_\partial|}dl_{\Sigma_\partial}\right)dl_{\Sigma_R} && (1.4)
\end{aligned}
$$

$dP_{\Sigma,GO}$, the portion as calculated via Geometrical Optics, is given by:

$$
dP_{\Sigma,GO} = \mathbb{1}_{\Sigma_\partial}(\vec{r}_{\text{spec}})\frac{P_0}{2\pi}\left(\frac{|\Gamma|^2\cos\theta_{\text{spec}}}{|\vec{r}_{\text{spec}}-\vec{r}_T|+|\vec{r}_R-\vec{r}_{\text{spec}}|}\right)dl_{\Sigma_R} \tag{1.5}
$$

1.1 Setup Simplifications

Assume that the second medium is a PEC strip starting at $(w_{x0}, 0)$ and ending at $(w_{xe}, 0)$:

$$
\Sigma_\partial = \{(w_x, 0) \in \mathbb{R}^2 \mid w_x \in [w_{x0}, w_{xe}]\} \tag{1.6}
$$

So that

$$
\vec{r}_{\text{spec}} = (w_{\text{spec}}, 0) \tag{1.7}
$$

The line of receivers is parallel to the strip:

$$
\Sigma_R = \{(r_x, r_y) \in \mathbb{R}^2 \mid r_x \in [r_{x0}, r_{xe}]\} \tag{1.8}
$$

The transmitter is fixed:

$$
\vec{r}_T = (t_x, t_y) \tag{1.9}
$$

We also assume that the source is linearly polarized with the electric field normal to the plane of incidence, so that

$$
|\Gamma|^2 = 1 \tag{1.10}
$$

We then get:

$$
\begin{aligned}
dx &= dl_{\Sigma_\partial} = dl_{\Sigma_R} \\
\frac{\cos\theta_{\Sigma_R}}{|\vec{r}_\partial - \vec{r}_T|} &= \frac{t_y}{t_y^2 + (w_x - t_x)^2} \\
\frac{\cos\theta_{\Sigma_\partial}}{|\vec{r}_R - \vec{r}_\partial|} &= \frac{r_y}{r_y^2 + (r_x - w_x)^2} \\
w_{\text{spec}} &= t_x + t_y \left(\frac{r_x - t_x}{r_y + t_y}\right) \\
&= r_x - r_y \left(\frac{r_x - t_x}{r_y + t_y}\right) \\
\implies \cos\theta_{\text{spec}} &= \frac{r_y + t_y}{\sqrt{(r_x - t_x)^2 + (r_y + t_y)^2}}, \quad \text{and} \\
|\vec{r}_{\text{spec}} - \vec{r}_T| + |\vec{r}_R - \vec{r}_{\text{spec}}| &= \sqrt{(r_x - t_x)^2 + (r_y + t_y)^2}
\end{aligned}
$$

which lead to the explicit spatial power distributions along the line of receivers:

$$
dP_{\Sigma_{R,S}} = \frac{P_0}{4\pi} \left(\int_{w_{x0}}^{w_{xe}} \frac{t_y r_y}{\left(t_y^2 + (x' - t_x)^2\right)\left(r_y^2 + (x - x')^2\right)} dx' \right) dx \tag{1.11}
$$

$$
dP_{\Sigma_{R,GO}} = \frac{P_0}{4\pi} \begin{cases} 2 \cdot \dfrac{t_y + r_y}{(x - t_x)^2 + (r_y + t_y)^2} dx & \text{if } w_{x0} \leq w_{\text{spec}} \leq w_{xe} \\ 0 & \text{otherwise} \end{cases} \tag{1.12}
$$

1.2 Example Setup

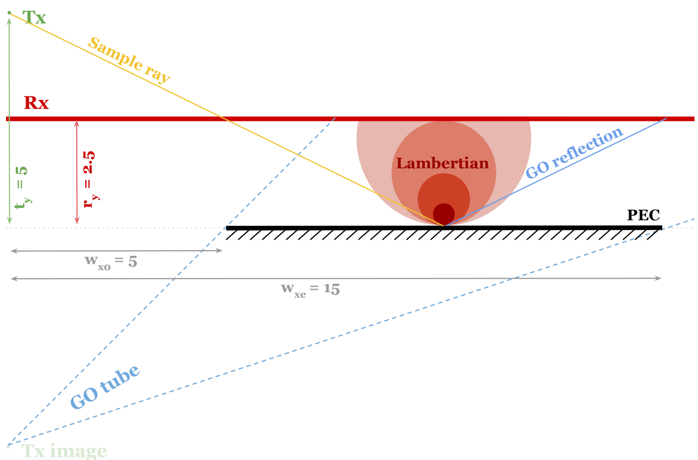

Figure 1: Line of receivers and fixed point source for a PEC strip of fixed width

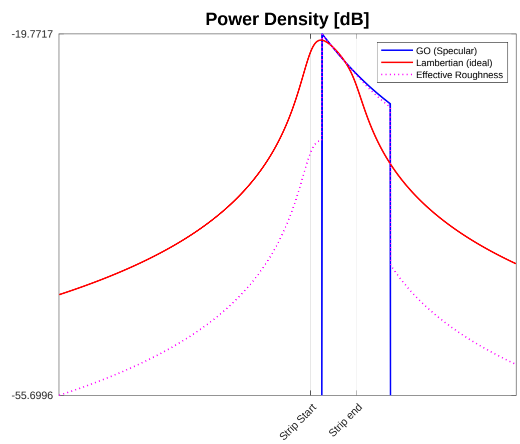

Figure 2: Power Density Profile for setup in Figure 1.

1.3 Lambertian Asymptotic

A virtue of implementing the ER model is the convergence of (1.11) under small values of the numerical parameter N_{strip}, the number of points along the strip for the integration, which is likely a consequence of the simple surface profile.

An even further simplification can be made, if we consider that the integrand in (1.11) is the product of two Cauchy distributions, and the whole integral, in the limit of an infinite wall, is the convolution of two Cauchy distributions. Under this limit, we can simplify (1.11) to

$$dP_{\Sigma_{R,S}} \sim \frac{P_0}{4} \cdot \frac{t_y + r_y}{(x - t_x)^2 + (t_y + r_y)^2}. \tag{1.13}$$

A comparison of (1.11) and (1.13) is shown in Figure 3 (limiting case).

1.4 Validation vs. Physical Optics Approximation

It's instructive to compare (1.11) to the physical optics case. Here, the incident electric field is given by a Hankel function:

$$E^i(\vec{r}) = E_0 H_0^{(2)}(k|\vec{r} - \vec{r}_T|)\hat{\mathbf{e}}_z \tag{1.14}$$

The incident magnetic field can be calculated from Maxwell's equations:

$$\begin{aligned}
H^i(\vec{r}) &= \frac{j}{\omega\mu_0} \nabla \times E^i(\vec{r}) \\
&= \frac{j}{\omega\mu_0} \begin{vmatrix} \partial_x & \partial_y & \partial_z \\ 0 & 0 & E_0 H_0^{(2)}(k|\vec{r} - \vec{r}_T|) \\ \hat{\mathbf{e}}_x & \hat{\mathbf{e}}_y & \hat{\mathbf{e}}_z \end{vmatrix} \\
&= \frac{jE_0}{\omega\mu_0} \left(\left(\partial_y (H_0^{(2)}(k|\vec{r} - \vec{r}_T|)) \right) \hat{\mathbf{e}}_x - \left(\partial_x (H_0^{(2)}(k|\vec{r} - \vec{r}_T|)) \right) \hat{\mathbf{e}}_y \right) \\
&= -\frac{jkE_0 H_1^{(2)}(k|\vec{r} - \vec{r}_T|)}{\omega\mu_0 |\vec{r} - \vec{r}_T|} \left((y - t_y)\hat{\mathbf{e}}_x - (x - t_x)\hat{\mathbf{e}}_y \right)
\end{aligned} \tag{1.15}$$

The Physical Optics surface current is then given by

$$\begin{aligned}
J^{PO}(\vec{r}) &= 2\hat{n} \times H^i(\vec{r}) = 2\hat{\mathbf{e}}_y \times H^i(\vec{r}) \\
&= \frac{2jkE_0(y - t_y) H_1^{(2)}(k|\vec{r} - \vec{r}_T|)}{\omega\mu_0 |\vec{r} - \vec{r}_T|} \hat{\mathbf{e}}_z
\end{aligned} \tag{1.16}$$

In our case $\vec{r} \in \{\ (x, 0) \mid x \in (w_{x0}, w_{xe})\ \}$, so $y = 0$. Using this, we can find the scattered electric field:

$$\begin{aligned}
E_S^{PO}(\vec{r}) &= \frac{k\eta_0}{4} \int_{\Sigma_\partial} J^{PO}(\vec{r}\,') H_0^{(2)}(k|\vec{r} - \vec{r}\,'|) dl' \\
&= -\frac{jkE_0 t_y}{2} \int_{\Sigma_\partial} \frac{H_1^{(2)}(k|\vec{r}\,' - \vec{r}_T|)\ H_0^{(2)}(k|\vec{r} - \vec{r}\,'|)\hat{\mathbf{e}}_z}{|\vec{r}\,' - \vec{r}_T|} dx'
\end{aligned} \tag{1.17}$$

Now, compute the scattered magnetic field (complex-conjugate), again from Maxwell:

$$\begin{aligned}
H_S^{PO*} &= \frac{-j}{\omega\mu_0}\left(\nabla \times E_S^{PO}\right)^* \\
&= -\frac{E_0 t_y}{2\eta_0}\left(\int_{\Sigma_\partial} \frac{H_1^{(2)}(k|\vec{r}\,' - \vec{r}_T|)\ \nabla \times (H_0^{(2)}(k|\vec{r} - \vec{r}\,'|)\hat{\mathbf{e}}_z)}{|\vec{r}\,' - \vec{r}_T|} dx'\right)^* \\
&= -\frac{kE_0 t_y}{2\eta_0}\left(\int_{\Sigma_\partial} \frac{H_1^{(2)}(k|\vec{r}\,' - \vec{r}_T|)H_1^{(2)}(k|\vec{r} - \vec{r}\,'|)(r_y\hat{\mathbf{e}}_x + ((x' - x)\hat{\mathbf{e}}_y))}{|\vec{r}\,' - \vec{r}_T||\vec{r} - \vec{r}\,'|} dx'\right)^* \\
&= -\frac{kE_0 t_y}{2\eta_0}\int_{\Sigma_\partial} \frac{H_1^{(1)}(k|\vec{r}\,' - \vec{r}_T|)H_1^{(1)}(k|\vec{r} - \vec{r}\,'|)(r_y\hat{\mathbf{e}}_x + ((x' - x)\hat{\mathbf{e}}_y))}{|\vec{r}\,' - \vec{r}_T||\vec{r} - \vec{r}\,'|} dx'
\end{aligned}$$

From this, we get

$$\begin{aligned}
dP_S^{PO} &= \frac{1}{2}\hat{\mathbf{n}} \cdot \mathrm{Re}(E_S^{PO} \times H_S^{PO*}) \\
&= -\frac{k^2 P_0 t_y^2 r_y}{4} \\
&\mathrm{Im}\left(\int_{\Sigma_\partial} \frac{H_1^{(1)}(k|\vec{r}_{T,\partial}|)H_1^{(1)}(k|\vec{r}_{\partial,R}|)}{|\vec{r}_{T,\partial}||\vec{r}_{\partial,R}|} dx' \int_{\Sigma_\partial} \frac{H_1^{(2)}(k|\vec{r}_{T,\partial}|)\ H_0^{(2)}(k|\vec{r}_{\partial,R}|)\hat{\mathbf{e}}_z}{|\vec{r}_{T,\partial}|} dx'\right) \qquad (1.18)
\end{aligned}$$

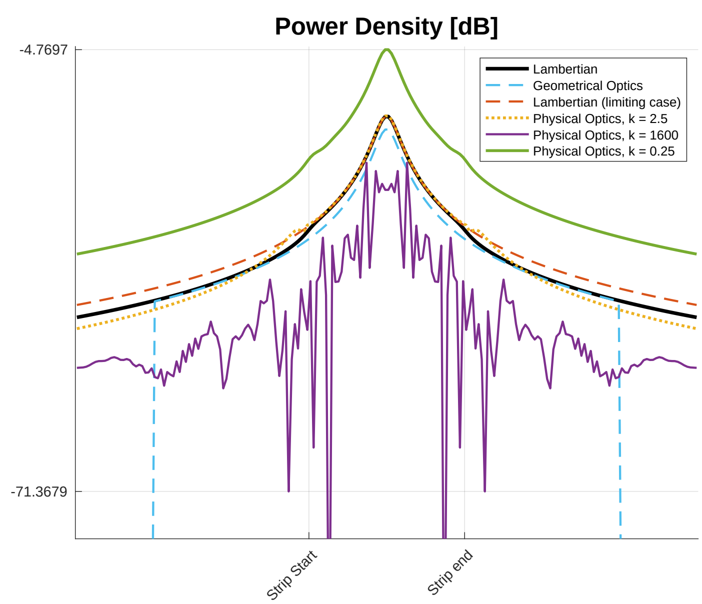

Figure 3: Model Validation vs. Physical Optics Approximation

Figure 3 is a revealing comparison and validation of the models seen so far for the simple setup. Among other things, it shows that:

- There is an inherent overestimation in the Lambertian models, independent of the scattering parameter S,
- This overestimation would likely carry over to directional models,
- So, new parametric models could certainly improve the accuracy and computational efficiency of the legacy models.

1.5 Power Delay Profile

We can also derive temporal power distributions (taking care with t-labels) from (1.11) and (1.12) by converting from spatial x, x' coordinates to temporal t, t' coordinates, via

$$ct = \sqrt{t_y^2 + (x - t_x)^2} + \sqrt{r_y^2 + (r_x - x)^2}, \quad \text{and} \tag{1.19}$$

$$ct' = \sqrt{t_y^2 + (x' - t_x)^2} \tag{1.20}$$

Inverting (1.20) for x' is a simple algebraic manipulation, whereas (1.19) inversion for x requires a change of coordinates so that the origin is at the specular point of reflection. Upon squaring twice, the x^4 and x^3 terms then cancel.

1.6 MATLAB Code

```
% Assume a cylindrical wave incident field
% The models used to compare are:
% 1. Effective Roughness (ER) Lambertian model
%    (overall ER contribution is an integral along the points of the wall), and
% 2. Geometrical Optics  (GO).
%
% Compare the time-averaged power density for the setup:
%
%     (r_x0, r_y)              (r_x, r_y)          (r_xe, r_y)
%                                                 /
%               (t_x, t_y)                     /
%                    \                       /
%                     \                   /
%                      \               /
%                       \          /
%                        \      /
%                         \/
% (w_x0,0)        (w_x, 0)            (w_xe, 0)
%
function dP = GO_Power_Density( x, ...              % Position along receiver line
                                P0, ...             % Source power
                                r_y, ...            % Rx antenna height
                                t_x, t_y, ...       % Tx antenna positions
                                w_x0, w_xe )        % Start and end of strip
% Formula 1.12 in derivations
   w_spec = t_x + t_y * ( (x - t_x)/(r_y + t_y) );
   if( w_x0 > w_spec || w_spec > w_xe )
```

```
      dP = 0;
   else
      dP = P0 * ( r_y + t_y ) / ...
               ( 2 * pi * ((x - t_x)^2 + (r_y + t_y)^2) );
   end
end

function dP = Lambertian_Power_Density( x, ...
                                        P0, ...
                                        r_y, ...
                                        t_x, t_y, ...
                                        w_x0, w_xe, ...
                                        N_strip )   % # Points along strip
% Formula 1.11 in derivations
   w_len = w_xe - w_x0;
   dx_w = w_len / N_strip;
   dP = 0;
   for( N = 1:N_strip )
      w_x = w_x0 + (N-0.5)*dx_w;
      numerator = t_y * r_y * dx_w;
      denominator = ((t_y)^2 + (w_x - t_x)^2) * ((r_y)^2 + (x - w_x)^2);
      ddP = ( numerator / denominator );
      dP = dP + ddP;
   end
   dP = dP * P0 / (4 * pi);
end

function dP = PO_Power_Density( x, ...
                                P0, ...
                                k, ...
                                r_y, ...
                                t_x, t_y, ...
                                w_x0, w_xe, ...
                                N_strip )   % # Points along strip
% Formula 1.20 in derivations
   w_len = w_xe - w_x0;
   dx_w = w_len / N_strip;
   int1 = 0;
   int2 = 0;
   for( N=1:N_strip )
      w_x = w_x0 + (N-0.5)*dx_w;
      r_TB = sqrt( (w_x - t_x)^2 + (t_y)^2 );
      r_BR = sqrt( (x - w_x)^2 + (r_y)^2 );
      num1 = besselh(1, 1, k*r_TB) * besselh(1, 1, k*r_BR );
      denom1 = r_TB * r_BR;
      num2 = besselh(1, 2, k*r_TB) * besselh( 0, 2, k*r_BR );
      denom2 = r_TB;
      int1 = int1 + dx_w*(num1/denom1);
      int2 = int2 + dx_w*(num2/denom2);
   end
   dP = imag( int1 * int2 );
   dP = - dP * (k*t_y)^2 * P0 * (r_y) / 4.0;
end

function dP = Lambertian_Power_Density_PO( x, ...
                                           P0, ...
                                           r_y, ...
                                           t_x, t_y, ...
                                           w_x0, w_xe )
% Formula 1.13 in derivations
    dP = (t_y + r_y)/((x - t_x)^2 + (t_y + r_y)^2);
    dP = P0 * dP / 4;%(4*pi);
end

%---Special Parameter Markings-------------------------------------------------------
%   [!] Interesting
%   [n] Numerical
%-----------------------------------------------------------------------------------
```

```
clc
clear all
close all
% Rx antennas------------------------------------------------------------------------
r_x0 = -200;                              % [!] First Rx
r_xe = 200;                               % [!] Last Rx
r_y = 5;                                  % [!] Rx height wrt strip
N_rx  = 200;                              % [n] Number of receivers
r_spread = r_xe - r_x0;                   %     Rx spread
dx_r = r_spread / N_rx;                   %     dx along Rx-line
r_x = zeros(1, N_rx);                     %     Rx positions
for( N = 1:N_rx )
   r_x( N ) = r_x0 + (N-0.5) * dx_r;
end
% Tx antenna-------------------------------------------------------------------------
t_x = 0;                                  %     Tx position
t_y = 2.5;                                % [!] Tx height wrt strip
% Strip------------------------------------------------------------------------------
w_x0 = -50;                               % [!] Strip start
w_xe = 50;                                % [!] Strip end
N_strip = 600;                            % [n] # Calc points along strip
w_len = w_xe - w_x0;                      %     Strip length
% Source waves-----------------------------------------------------------------------
P0 = 1;                                   %     Source power
k_sweet = 2.5e0;                          % [!] Wavenumber, sweetspot
k_low = 0.1 * k_sweet;                    % [>] Wavenumber, low
k_high = 1600;                            % [!] Wavenumber, high (EHF)
% Models-----------------------------------------------------------------------------
S = sqrt(0.1);                            % [!] Effective Roughness
R = sqrt( 1.0 - (S)^2);                   %     Specular reflectance reduction

Lambertian_Density = zeros(1, N_rx);
Lambertian_Density_PO = zeros(1,N_rx);
PO_Density_Sweet = zeros(1,N_rx);
PO_Density_High = zeros(1,N_rx);
PO_Density_Low = zeros(1,N_rx);
GO_Density = zeros(1, N_rx);
ER_Density = zeros(1, N_rx);
r_spec_start = NaN; %prealloc
r_spec_end = NaN; %prealloc
for(N_r = 1:N_rx)
   x = r_x( N_r );
   this_w_spec = t_x + t_y * ( (x - t_x)/(r_y + t_y) );
   if( w_x0  <= this_w_spec && this_w_spec <= w_xe && isnan(r_spec_start) )
       r_spec_start = x;
   elseif( w_xe < this_w_spec && isnan(r_spec_end) )
       r_spec_end = x;
   end
   PO_Density_Sweet( N_r ) = PO_Power_Density( x, ...
                                               P0, k_sweet, ...
                                               r_y, ...
                                               t_x, t_y, ...
                                               w_x0, w_xe, ...
                                               N_strip );   % # Points along strip
   PO_Density_High( N_r ) = PO_Power_Density( x, ...
                                              P0, k_high, ...
                                              r_y, ...
                                              t_x, t_y, ...
                                              w_x0, w_xe, ...
                                              N_strip );   % # Points along strip
   PO_Density_Low( N_r ) = PO_Power_Density( x, ...
                                             P0, k_low, ...
                                             r_y, ...
                                             t_x, t_y, ...
                                             w_x0, w_xe, ...
                                             N_strip );   % # Points along strip
   GO_Density( N_r ) = GO_Power_Density( x, ...
                                         P0, r_y, t_x, t_y, ...
```

```
                                        w_x0, w_xe );
    Lambertian_Density( N_r ) = Lambertian_Power_Density( x, ...
                                                         P0, r_y, t_x, t_y, ...
                                                         w_x0, w_xe, ...
                                                         N_strip );
    ER_Density( N_r ) = (S^2) * Lambertian_Density( N_r ) + (R^2) * GO_Density( N_r );
    Lambertian_Density_PO( N_r ) = Lambertian_Power_Density_PO( x, ...
                                                               P0, r_y, t_x, t_y, ...
                                                               w_x0, w_xe );
    % Use decibels:
    GO_Density( N_r ) = 10 * log10( GO_Density( N_r ));
    Lambertian_Density( N_r ) = 10 * log10( Lambertian_Density( N_r ));
    ER_Density( N_r ) = 10 * log10( ER_Density( N_r ));
    Lambertian_Density_PO( N_r ) = 10 * log10( Lambertian_Density_PO( N_r ));
    PO_Density_Sweet( N_r ) = 10 * log10( PO_Density_Sweet( N_r ));
    PO_Density_High( N_r ) = 10 * log10( PO_Density_High( N_r ));
    PO_Density_Low( N_r ) = 10 * log10( PO_Density_Low( N_r ));
end
% handle -inf for dB graphs
for(N_r = 1:N_rx)
    % replace complex values with -inf
    if( imag(PO_Density_Sweet(N_r)) > 1e-12 )
        PO_Density_Sweet(N_r) = -inf;
    else
        PO_Density_Sweet(N_r) = real(PO_Density_Sweet(N_r));
    end
    if( imag(PO_Density_High(N_r)) > 1e-12 )
        PO_Density_High(N_r) = -inf;
    else
        PO_Density_High(N_r) = real(PO_Density_High(N_r));
    end
    if( imag(PO_Density_Low(N_r)) > 1e-12 )
        PO_Density_Low(N_r) = -inf;
    else
        PO_Density_Low(N_r) = real(PO_Density_Low(N_r));
    end
end
all_data = [ GO_Density(:), Lambertian_Density(:), ER_Density(:), ...
             PO_Density_Sweet(:), PO_Density_High(:), PO_Density_Low(:) ];
finite_values = all_data(isfinite(all_data));
min_finite_val = min( finite_values );
max_finite_val = max( finite_values );
inf_display_val = min_finite_val - 0.5 * ( max_finite_val - min_finite_val );
for(N_r = 1:N_rx)
    if( GO_Density(N_r) == -inf )
        GO_Density(N_r) = inf_display_val;
    end
    if( Lambertian_Density(N_r) == -inf )
        Lambertian_Density(N_r) = inf_display_val;
    end
    if( ER_Density(N_r) == -inf )
        ER_Density( N_r ) = inf_display_val;
    end
    if( PO_Density_Sweet(N_r) == -inf )
        PO_Density_Sweet( N_r ) = inf_display_val;
    end
    if( PO_Density_High(N_r) == -inf )
        PO_Density_High( N_r ) = inf_display_val;
    end
    if( PO_Density_Low(N_r) == -inf )
        PO_Density_Low( N_r ) = inf_display_val;
    end
end

% default color ordering
new_colors = colororder;
figure;
%plot( r_x, GO_Density, ...
```

```
%       'b', ...
%       'LineWidth', 1.5, ...
%       'DisplayName', 'GO (Specular)' );
hold on;
plot( r_x, Lambertian_Density, ...
      'Color', 'k', ...
      'LineWidth', 2.5, ...
      'DisplayName', 'Lambertian' );
plot( r_x, GO_Density, '--', ...
      'Color', new_colors(6,:), ...
      'LineWidth', 1.5, ...
      'DisplayName', "Geometrical Optics" );
plot( r_x, Lambertian_Density_PO, '--', ...
      'Color', new_colors(2,:), ...
      'LineWidth', 1.5, ...
      'DisplayName', 'Lambertian (limiting case)' );
plot( r_x, PO_Density_Sweet, ':', ...
      'Color', new_colors(3,:), ...
      'LineWidth',2.0, ...
      'DisplayName', "Physical Optics, k = " + k_sweet );
plot( r_x, PO_Density_High, ...
      'Color', new_colors(4,:), ...
      'LineWidth', 1.5, ...
      'DisplayName', "Physical Optics, k = " + k_high );
plot( r_x, PO_Density_Low, ...
      'Color', new_colors(5,:), ...
      'LineWidth', 2.0, ...
      'DisplayName', "Physical Optics, k = " + k_low );
grid on;
ylim([min_finite_val*1.1, max_finite_val*0.9]);
title_string = "Power Density [dB]"
title( title_string, ...
       'FontSize', 17, ...
       'FontWeight', 'bold' )
ax = gca;
special_locations = [ w_x0, w_xe ];
special_labels = { 'Strip Start',  'Strip end' };
[ sorted_locations, sort_order] = sort(special_locations);
sorted_labels = special_labels(sort_order);
ax.XTick = sorted_locations;
ax.XTickLabel = sorted_labels;
ax.XTickLabelRotation = 45;
ax.YTick = [ min_finite_val, max_finite_val ];
hold off;
legend;
```